AMENDMENTS TO THE SPECIFICATION

On page 1, please replace the subheading "FIELD OF THE INVENTION" with "FIELD".

On page 1, please replace the subheading "BACKGROUND OF THE INVENTION" with "BACKGROUND".

On page 3, please replace the subheading "SUMMARY OF THE INVENTION" with "SUMMARY".

On page 9, please replace the subheading "DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS" with "DETAILED DESCRIPTION OF THE VARIOUS EMBODIMENTS".

Applicants note the original application was filed without paragraph numbers. To facilitate the incorporation of these amendments, Applicants have numbered the paragraphs. Please replace Paragraphs [0001], [0002], [0007-0008], [0011-0015], [0017-0019], [0021], [0058-0060], [0069], [0074], [0078-0079], [0128], [0164], [0169-0171], [0173-0174] with the following paragraph rewritten in amendment format:

[0001] This disclosure was made with Government support under Contract Number F33615-00-C-6061 awarded by the Air Force Research Laboratory. The Government has certain rights in this ~~invention~~disclosure.

[0002] The field of the present ~~invention~~disclosure relates to the communication of data over a data bus that interconnects a plurality of data processors, particularly data processors residing on different physical boards.

[0007] Furthermore, many multi-board processing systems are implemented in a manner that makes space a premium commodity. In one application of the present ~~invention~~disclosure, a strike helmet for pilots such as the Strike Helmet 21 project by the assignee of the present ~~invention~~disclosure, the data processing boards are seated in a Versa Module Europa (VME) chassis such that an insufficient number of hardware slots are available for a variety of communication methods (such as Fibre Channel). In cases such as this, in addition to providing flexibility for evolving communication frameworks, the implementation of a multi-board communication utility should also provide space efficiency to satisfy narrow size constraints.

[0008] Having been unable to find an existing communication utility that satisfies some or all of these needs in the art, the inventors herein developed the present ~~invention~~disclosure.

[0011] Moreover, it is preferred that at least one channel, and more preferably each channel, be user-redefinable with any of a plurality of available configuration types. Examples of available configuration types for the present ~~invention~~disclosure include: (1) a copy on send configuration type, (2) a copy to pool on receive configuration type, (3) a copy to buffer on receive configuration type, (4) a push to pool on receive configuration type, (5) a push to buffer on receive configuration type, (6) a queue on send configuration type, (7) a copy to self configuration type, (8) a queue to self configuration type, and (9) an overwrite on send configuration type.

[0012] According to another aspect of the present ~~invention~~disclosure, disclosed herein is a data processing apparatus comprising: (1) a first data processing

board; (2) a second data processing board; (3) a bus connecting the boards with each other; and wherein each board comprises a communication utility for communicating data over the bus to the other board, and wherein the communication utility communicates data according to a redefinable configuration such that a bus utilization percentage in a range of at least 13% for 8 Kbyte transfers is achieved. This bus utilization percentage is measured from the time that the sending board calls vcuSend() to the time that the receiving board returns from vcuRecv() (that is, makes the data available to the application). Also, this bus utilization was achieved without the boards' cache snooping being enabled. When using a board with cache snooping capabilities, it is expected that a bus utilization of approximately 25% for 8 Kbyte transfers can be reached. With other known communication utilities, such as TCP/IP over a VME bus, the bus utilization percentage is much lower, around 5% for 8Kbyte transfers.

[0013] According to another aspect of the present ~~invention~~disclosure, disclosed herein is a method of configuring a communication utility for transporting data from a first processor to a second processor over a bus, the method comprising: (1) defining a configuration for a channel through which data is communicated over a bus by a communication utility interfacing at least a first processor with a second processor; and (2) in accordance with the defined channel configurations, compiling software for controlling the communication utility.

[0014] By encapsulating the configuration of the system, a developer is relieved of the need to be aware of the system's topology and channel transmission characteristics. With the present ~~invention~~disclosure, it is preferred that the system topology and channel transmission characteristics be set at the configuration level.

Thus, the developer's task is made easier (1) because of the system's flexibility, and (2) because the differences between inter-board and intra-board communication and the differences between channel transmission characteristics are configured separately from the developer's software. That is, the topology and transmission characteristics of the channel(s) exist separately from the application(s) using the channel(s). Therefore, because they are not interwoven, a change to a channel does not require a change to the application using that channel. Further still, because the memory is not dynamically allocated in the present ~~invention~~disclosure, delays attributable to such dynamism are not present. Yet the present ~~invention~~disclosure memory allocation is still capable of retaining efficiency due to the flexible nature of its user-configurability.

[0015] The present ~~invention~~disclosure may also provide a user interface for configuring each channel separately from the application software. That is, the application(s) using the present ~~invention~~disclosure to communicate data need not be cognizant of the configurations of the various communication channels. Thus, according to yet another aspect of the present ~~invention~~disclosure, disclosed herein is a device comprising: (1) a user interface through which a user provides configuration data; and (2) a processor configured to receive the configuration data from the user interface and generate a configuration file therefrom, the configuration file comprising configuration information for a plurality of channels over a bus that interconnects a plurality of data processing boards.

[0017] Further still, according to yet another aspect of the present ~~invention~~disclosure, disclosed herein is a device comprising: (1) a user interface

through which a user specifies a stored configuration file, the configuration file comprising configuration information for a plurality of channels over a bus that interconnects a plurality of data processing boards; and (2) a processor configured to retrieve the specified configuration file and generate software in accordance with the retrieved configuration file, the software for controlling data communications over the bus between the boards. Here, the user interface is a UNIX command line interface.

[0018] The software aspects of the present ~~invention~~disclosure can be implemented on any form of computer-readable media, including but not limited to compact disks, floppy disks, processor memory, a network-accessible server, and the like.

[0019] Preliminary testing of a prototype of the present ~~invention~~disclosure indicates that the present ~~invention~~disclosure performs better than current communication utilities available in the art. These and other features and advantages of the present ~~invention~~disclosure will be in part pointed out and in part apparent upon review of the following description, figures, and claims.

[0021] Figure 2 illustrates an exploded block diagram of a preferred communication utility for the present ~~invention~~disclosure;

[0058] Figures 36-47 are comparative data charts indicating the performance of various configurations of the preferred embodiment of the present ~~invention~~disclosure relative to other communication utilities.

[0059] *System Overview:* Figure 1 illustrates ~~a preferred~~ an embodiment of the present ~~invention~~disclosure. In Figure 1, a multi-board data processing system 100 comprises a first data processing board 102 and second data processing board 104, wherein the two boards are interconnected via a bus 110. Each board has one or more data processing applications 108 running thereon. When data is to be transferred from one application to another (whether an interboard transfer or an intraboard transfer), the communication utility 106 resident on each board is used. The communication utilities 106 interface each board with one another via the bus 110. The data transferred over the bus can be of either a fixed size or a variable size. The communication utility 106 communicates data such that a bus utilization percentage is in a range from approximately 13% to approximately 25% for 8 Kbyte data transfers across the bus 110.

[0060] It is preferred that the data bus 110 be a Versa Module Europa (VME) bus, and that the boards be VME boards. In particular, it is preferred that the present ~~invention~~disclosure use the Dy4 family of VME boards, such as the Dy4 179, 181, and 712 boards, which are publicly available from Force Computers, Inc. However, as would be understood by those of ordinary skill in the art, the system 100 can be implemented with data processing boards other than VME boards, including but not limited to PCI boards on which mailboxes and DMA can be implemented through either hardware or software, similar ISA boards, or any board types with parallel back planes and on which mailboxes and DMA can be configured through either hardware or software. However, VME boards are preferred because the inventors herein have found them to be more easily configurable with respect to mailboxes and DMA. Also, it is worth noting that while two boards are depicted in the system of Figure 1, the present

~~invention~~disclosure is capable of supporting more than two boards communicating with each other over the bus, and further as will be explained in more detail below, this number can be user-definable.

[0069] The preferred embodiment of the present ~~invention~~disclosure preferably uses three variations of the vcuSend() routine, with the particular vcuSend() routine being used depending upon the channel configuration of the channel involved in the data transfer. However, it should be noted that a vcuSend() routine that requires more information than another vcuSend() routine can preferably be substituted for that other vcuSend() routine.

[0074] To receive data, the preferred embodiment of the present ~~invention~~disclosure preferably uses the vcuRecv() routine: int vcuRecv(int *channel*, char* *&data*, int *&dataSize*, int *flags*). The vcuRecv() routine operates to receive the next message waiting in the Rx queue for the channel specified in the argument, if such a message exists. The *flag* options for vcuRecv() are VCU_NO_BLOCK (which is the default setting) and VCU_BLOCK. When the flag is VCU_BLOCK, the vcuRecv() routine blocks until the data arrives. However, it is worth noting that a timeout option can be used to end the block after the passage of a specified amount of time.

[0078] *Channel Configuration Types:* The present ~~invention~~disclosure preferably allows user to define (and redefine) the transmission characteristics of at least one communication channel, and more preferably, each communication channel.

It is preferred that the user be given the ability to define (and redefine) aspects such as: the number of communication channels, the maximum size of a single data transfer for each channel, the conditions under which DMA is used for data transfers across the bus, and how each channel is to handle data transfers.

[0079] In ~~the preferred embodiment~~an implementation of the present ~~invention~~disclosure, the user is provided with a plurality of selectable configuration types which include a variety of different settings for these aspects in a single package. The preferred configuration types for the present ~~invention~~disclosure are: (1) "copy on send", (2) "copy to pool on receive", (3) "copy to buffer on receive", (4) "push to pool on receive", (5) "push to buffer on receive", (6) "queue on send", (7) "copy to self", and (8) "overwrite on send". Figure 5 is a table that provides a description of how each configuration type can be handled on the sending side and receiving side. It is worth noting that it is preferable to use the sender side sequence for push to pool on receive for send calls with all configurations. Similarly, it is preferable to use the receiver side sequence for either copy to buffer on receive or push to pool on receive for receive calls with all configurations.

[0128] Through fields 256, 258, 260, and 262 the user can define the size of the channel's receive queue, receive pool, transmit pool, and push queue respectively. Figures 28 and 29 describe the preferred setting for these sizes by configuration type. These preferred values assume worst-case application use, and best-case internal performance. If the system is so heavily loaded with data transfers that it does not have time to process one send before another starts, additional buffer space may be

required. Because the transmit pool is especially dependent on the system load (a command to release a transmit pool slot comes from a remote board, so there is a delay between the command to release it and the actual release, wherein the delay is dependent upon the system load), its size should be given attention, and extra slots should be allocate thereto if the system is a heavily-loaded one. Also, while the values in Figures 28 and 29 are preferred values, it should be understood that practitioners of the ~~invention~~disclosure may choose to select values other than those shown in Figures 28 and 29.

[0164] The preferred system of the present ~~invention~~disclosure also preferably allows for multicasting in certain situations. Multicast is an efficient method of sending a message to multiple boards. With the VCU, multicast saves some processing time despite the bulk of the processing time coming when the data is moved across the VME bus, and when each board makes its own copy. The VCU multicast also makes it easier for a programmer to indicate multiple boards as well as saves memory space.

[0169] The storage class preferably stores all errors in a queue. Access to the front of the queue proceeds through the API call vcuErrno(). The routine vcuClearErrno() removes the errno at the front of the queue and returns it. The routine vcuLastErrno() returns the most recent errno, but does not clear it from the queue. The value in vcuErrno() can be read from other boards in the VCU system by a vcuRequestErrno() call, which takes the destination ID of the board as an argument. If vcuRequestErrno() is called as a multicast, the return value is zero for no errors and

VCU_ERROR_NO_CODE if any board reports an error. The vcuRequestErrno() routine does require some VCU communication to be working in order to report errors from other boards. If the storage class fills up, subsequent error messages are lost. It is preferred to make the storage class have a size sufficient for 10 messages, however, it is even more preferred to make the storage class have a size that is user-configurable. Figures 33(a) and (b) list and describe the preferred error messages of the present ~~invention~~disclosure, which are either return values from application calls to the VCU API routines or are error codes recorded inside the VcuSocket object. The values are all preferably defined in vcuDefines.h.

[0170] Thus, the present ~~invention~~disclosure represents a highly efficient communication utility for managing communications over a bus between a plurality of data processing boards. Testing has indicated that the present ~~invention's~~disclosure's efficient user-definable configurations lead to greatly improved performance relative to other known communication utilities. Figures 36-47 are exemplary of the results of such testing. Figure 36 illustrates transfer time (in microseconds) versus transfer size (in bytes) as measured for the present ~~invention~~disclosure under the "copy on send" configuration and other communication utilities using the TCP/IP protocol to communicate over the shared bus. As can be seen, the present ~~invention~~disclosure performs significantly better than the other techniques, particularly for larger transfer sizes. Figure 37 is a zoomed-in view of the smaller end of the transfer size axis for the chart of Figure 36. As can be seen, the present ~~invention~~disclosure also outperforms the other techniques for smaller transfer sizes. Figures 38-47 illustrate similar phenomena, relative to current communication techniques, for the present

~~invention~~disclosure under other configuration types such as "copy to buffer on receive", "copy to pool on receive", "copy to self", "push to buffer on receive", and "push to pool on receive".

[0171] While the present ~~invention~~disclosure has been described above in relation to its preferred embodiment, various modifications may be made thereto that still fall within the ~~invention~~disclosure's scope, as would be recognized by those of ordinary skill in the art.

[0173] Further, various routines can be added to the system, and the user-definability of various system parameters can be added or removed as desired by a practitioner of the ~~invention~~disclosure. Routines that can be added include a routine to clear all memory pools on the board (as a way of resetting the VCU), routines to create, destroy, send, and wait for multi-board events including possibly semaphores, a VcuLookup() to convert error numbers into descriptions, and a routine for printing of the CvcuDualBuffer class. Also, user-definability can be enhanced with the ability to define a maximum vcuEvent count to the configuration capabilities.

[0174] Such modifications to the ~~invention~~disclosure will be recognizable upon review of the teachings herein. As such, the full scope of the present ~~invention~~disclosure is to be defined solely by the appended claims and their legal equivalents.